

Real-time Multi-device Control System Implementation for Natural User Interactive Platform

Myoung-Jin Kim¹ Tae-min Hwang¹ Sung-Hun Chae¹ Min-Joon Kim¹ Yeon-Kug Moon^{1*} SeungJun Kim¹

ABSTRACT

Natural user interface (NUI) is used for the natural motion interface without using a specific device or tool like a mouse, keyboards, and pens. Recently, as non-contact sensor-based interaction technologies for recognizing human motion, gestures, voice, and gaze have been actively studied, an environment has been prepared that can provide more diverse contents based on various interaction methods compared to existing methods. However, as the number of sensors device is rapidly increasing, the system using a lot of sensors can suffer from a lack of computational resources. To address this problem, we proposed a real-time multi-device control system for natural interactive platform. In the proposed system, we classified two types of devices as the HC devices such as high-end commercial sensor and the LC devices such astraditional monitoring sensor with low-cost. we adopt each device manager to control efficiently. we demonstrate a proposed system works properly with user behavior such as gestures, motions, gazes, and voices.

✉ keyword: Natural user interface, Immersive experience, Sensor interface

1. Introduction

Natural user interface (NUI) is used for the natural motion interface without using a specific device or tool like a mouse, keyboards, and pens [1]. Recently, compulsory indoor living increase and also connectivity and relationship between spatial environments is interested with untact era, to build an interactive environment for providing realistic content services in various spaces, the natural user interface (NUI) platform technology keep developing with multi-channel cameras or various sensors to detecting human behavior or collecting spatial information and processing in real-time to providing users immediately.

In addition, changes in users' requirements for various content-based services in an indoor environment go beyond the simple consumption stage to obtain additional information of services by selecting and judging by themselves, and most users want direct participation.

In particular, in the case of user interaction-based service contents, as 3D stereoscopic media is rapidly progressing, the number of contents services that enable immersive experience is rapidly increasing. Accordingly, we will continue to demand services based on the NUI platform technology that recognize the user's behavior, gaze, and voice according to the user's intention, process it in real time and reflect it in the content. is becoming. In general, most interaction technologies real-time interaction centered on devices, processors and humans, is a very important factors whichis a very important part in the composition of the spatial interaction environment then the type and number of devices rather than the interaction spatial environment [2-3].

In order to obtain a user's desired action for interaction in space with conventional technology, a service is provided in the form of providing limited content in a predetermined pattern centered on a touch sensor device such as a touch method or a button method. Therefore, existing research has

¹ Dept. of Data Convergence Platform Research Center, Korea Electronics Technology Institute, Gyeonggi-do(Seongnam-si), 13509, South Korea

² School of Integrated Technology (RT, Intelligent Robotics), Gwangju Institute of Science and Technology, Gwangju, 61005, South Korea,

* Corresponding author: Yeon-kug Moon(ykmoon@keti.re.kr)

[Received 8 Octboer 2021, Reviewed 14 October 2021(R2 24 Novembmer 2021), Accepted 1 Decebmer 2021]

☆ This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project number: R2020040058)

☆ A preliminary version of this paper was presented at APIC-IST 2021.

been focused on simple interaction devices without limiting the input method, device, and display, which are concept of interaction, or defining space. Recently, as non-contact sensor-based interaction technologies for recognizing human motion, gestures, voice and gaze have been actively studied, an environment has been prepared that can provide more diverse contents based on various interaction methods compared to existing methods. For example, interaction is provided through an interface [4] that the user directly manipulates in the existing real world. With the development of various contact-type multi-interaction methods [5-6], spatial interaction techniques for motion recognition using various sensors [7-9] have been studied. Accordingly, there is an increasing demand for technology capable of simultaneously controlling and processing multi-channel cameras or multiple sensor devices in real time, but it is still limited to single device centered interaction related technologies. Basically, interaction elements exist centered on people in space and are based on real-time interaction through organic relationships between devices, processors, and people. Therefore, to provide real-time interaction in such a space, a new type of interaction system structure for selection and concentration of user information and multi-channel signal processing system for low-delay processing of multiple data are required.

In this paper, we propose an NUI platform technology for a real-time multi-device control system for a NUI platform, a multi-channel sensor device-based system architecture to provide user-customized interactions. Our proposed system considered two types of devices as reducing process overload due to amount of data processing. The heavy computation (HC) devices mainly process the video, audio and the other large amount data which is received by camera, commercial sensor, MIC, touch panel and so on. On the other hand, the low computation (LC) devices mainly handle multiple sensors, which deal low amount data, and also support various type of universally I/O to multi devices. We structurally integrated two systems and connected various sensors and devices to simulating the operation of integrated system with test graphic user interface (GUI).

The rest of this paper is organized as follows. Section II covers our proposed system. In Section III, we present the

implementation result by two type of system model of NUI such as Heavy Computation (HC) and Low Computation (LC) devices. Finally, Section IV concludes our work.

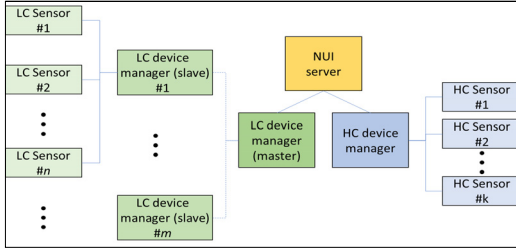
2. Multi-device Control System for Natural User Interactive Platform

2.1 Proposed System

In this paper, we consider two types of devices as the HC devices such as high-end commercial sensor like webcam or motion sensor and the LC devices such as traditional monitoring sensor across a wide area with low-cost [10]. In order to increase accuracy of the user behavior and recognize various user interaction, the system should control two types of devices at the same time. Thus, in the proposed system, we adopt a HC device manager and a LC device manager to control various sensor devices efficiently.

A HC device manager is considered to require large amounts of data from sensors for a high performance processing such as image processing, object detection, and deep learning. Thus, CPU/GPU embedded system is used for the HC device manager. On the other hand, a LC device manager is considered to wirelessly control multi-devices that require low computation, such as switches. The LC device manager is consist of master and slave board. The master board that controls the LC system and the slave board wirelessly connected are installed throughout the content space to provide a simple and intuitive interaction effect.

Lastly, NUI server gathers user interaction data from the HC device manager and the LC device manager. The connection between NUI server and the HC device manager works is on TCP/IP for high-speed data transfer, and connection between NUI server and the LC device manager is on Wi-Fi and BLE for wireless communication. Figure 1 shows a structure of the proposed multi-device control system for natural user interactive platform. We design the structure hierarchically to ensure robustness as a performance attribute in the proposed system.



(Figure 1) Structure of the proposed multi-device control system for natural user interactive platform

3. Implementation

3.1 HC Device Manager Implementation

This section describes the HC device manager. Various sensors are connected to HC device manager and the sensors gather information from user and transmit it to the HC device manager. The HC device manager recognizes user interaction with the gathered information and transfer it to the NUI server via middleware based on TCP/IP. If it is necessary to interact with multi-user, several HC device managers can be attached to the NUI server as shown in Figure 2. Figure 2 shows a system architecture of the HC device manager and NUI server.

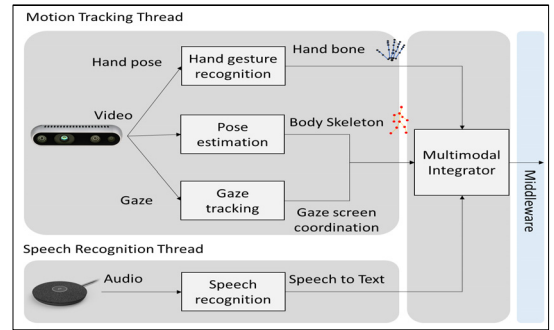
Compared to the LC device manager, the HC manager exploits the capabilities of a high performance CPU/GPU embedded system in order to estimate user interaction by CNNs (Convolution Neural Networks) [11-15]. In this paper,

we select NVIDIA Jetson AGX Xavier with 8-Core ARM 64bit CPU and 512-core Volta GPU with Tensor [16]. In this section, we describe the details of the sensors and middleware in the HC device manager respectively.

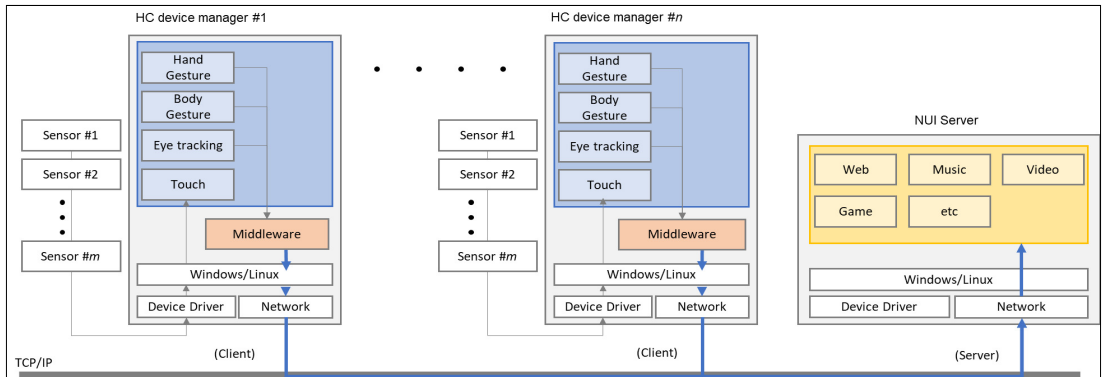
3.1.1 Sensors

We consider the system which supports pose estimation, hand gesture recognition, gaze tracking and speech recognition. Based on these requirements, we select the hardware as follows: Intel RealSense D435 for hand gesture recognition, pose estimation and gaze tracking, and logitech Rally MIC pod for speech recognition.

Figure 3 shows a block diagram of a HC device manager with the sensors. The HC device manager creates multiple threads to synchronize the data from the different sensors. Then, the multimodal integrator estimates user interaction by analyzing the sensor data.



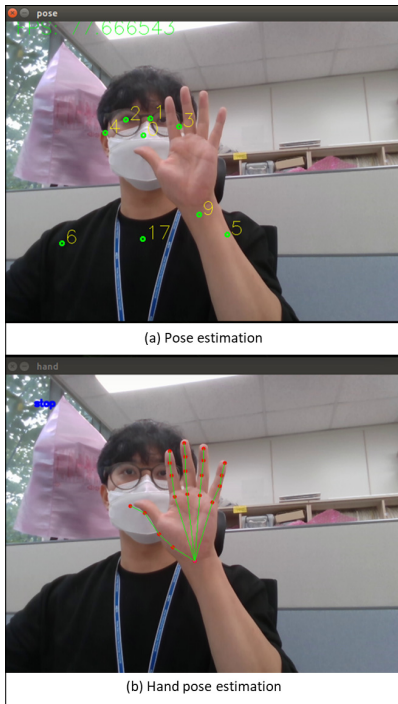
(Figure 3) Block diagram of HC device manager



(Figure 2) Architecture of the HC device manager and NUI server

For instance, In the hand gesture recognition, hand bone coordinates can be obtained including the original position and orientation of joints and fingertips including the specific hand gesture like stop sign, fist gesture, and peace gesture can be extracted. Also, in the pose estimation, body skeleton coordinates and posture of user can be obtained such as standing, sitting, lying down or doing some activity.

In this paper, we use a method for estimating pose and hand pose using Tensor RT pose (TRT Pose), which is an open-source NVIDIA project that aims to enable real-time pose estimation based on deep learning [11-12], for gaze tracking using MPII gaze tracking [17], and for speech recognition using Kakao speech API. Figure 4 shows an example of motion tracking with TRT pose. Figure 4(a) shows the method can find keypoints of pose estimation and display it on the screen. Likewise, Figure 4(b) shows classifying hand gesture. These pose estimation method and hand gesture method can work simultaneously with the same video source.



(Figure 4) Example of motion tracking.

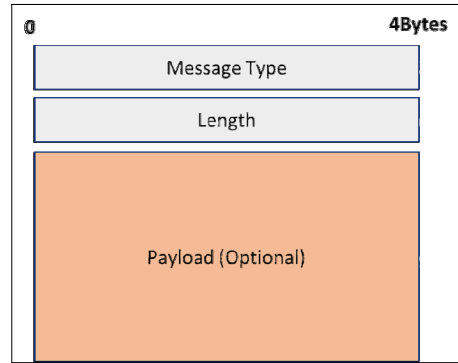
(a): Pose estimation, (b): hand pose estimation

3.1.2 Middleware

As we describe previously, the NUI system which uses a lot of sensors can suffer from a lack of computational resources like CPU or memory units. To address this problem, we physically separate the HC device managers that has several sensors from the NUI server. Instead of separating the sensors from NUI server, we adopt the middleware to integrate system.

In addition, middleware has a role of masking the distribution of the underlying sensor hardware by hiding low-level hardware details such as scalable service-oriented middleware on ethernet/IP (SOME/IP) and data distribution service (DDS) [18-19].

In the HC device manager, The middleware is based on TCP/IP for offering high speed data transmission. In this paper, we defined a simple protocol for the data transmission between the HC device manager and NUI server as shown in Figure 5. The packet format basically consists of *Message type*, *Payload length*, and *Payload*.



(Figure 5) Packet format for the data transmission

(Table 1) A table of message type field

Number	Value
0x00	Fire and forget
0x01	Subscription
0x02	Service request
0x03	Service response

A. Message type

The *Message type* field is used to differentiate different types of message as shown in Table 1. The field is based on the idea of the SOME/IP for the vehicle communication [18]. According to the communication patterns, the middleware of the HC device manager provides three types of the methods: Fire and forget (0x00), subscription (0x01), and service (0x02 and 0x03).

First, the middleware of the HC device manager provides Fire and forget type, which means request without response message. For example, when the user swipes their hand to the right, a sensor recognizes pose of user and sends data including *Message type* (0x00) to the NUI server without response from the NUI server. Figure 6(a) shows an example of the Fire and forget.

Secondly, a certain information should be cyclically transferred to the NUI server. For instance, the user's gaze or body skeleton can be periodically transmitted to the NUI server. In this case, the HC device manager send data including *Message type* (0x01). Figure 6(b) shows an example of subscription.

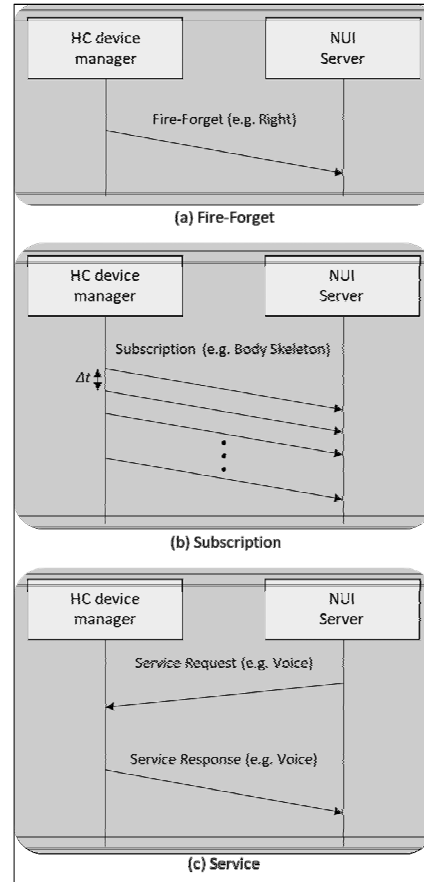
Finally, we can consider the case that the NUI server may call a service to the HC device manager to received the data. Figure 6(c) shows an example Service request and response. When the NUI server calls a service request (0x02) to convert speech to text, the HC device manager can listen user's speech and convert it to text. Then, it responds the converted text to the NUI server by sending the service response (0x03).

B. Length

*Length*field contains the length in Byte starting from itself until the end of the packet. Since TCP supports segmentation of payload, *Length* field should be included.

C. Payload

In the *Payload*field, the user interaction data can be carried. In this paper, we consider payload including user interaction such as screen x-y coordinates of the gaze, user gesture, text, and so on. Table 2 shows an example of the payload for gaze tracking.



(Figure 6) Sequence diagram of middleware according to the message type

(Table 2) An example of the payload for gaze tracking

Type	Value
int64	name (e.g. gaze)
int64	x coordinates of gaze
int64	y coordinates of gaze

3.2 LC Device Manager Implementation

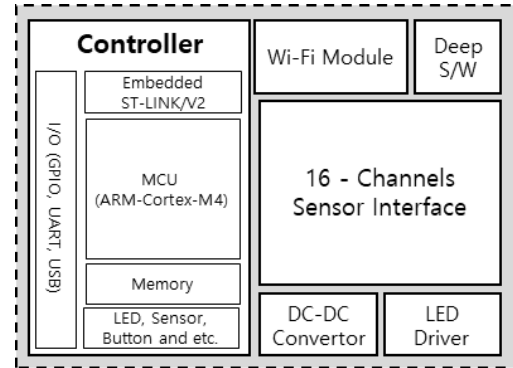
3.2.1 Master-Slave Control Board

The master-slave based wireless control system is designed to provide an immersive experience in a large space by low computation devices. The master board

controls the slave board by wireless communication, and the slave board controls the multi-sensor device. The structure of the LC control system is simply shown in Figure 7. The control board system is a control system capable of interworking and simultaneous control regardless of sensor types such as button switches, IR sensors, light devices, sound output devices. The wireless communication of the master board that controls the slave board includes a device control function for two-way feedback through user data collection from multi-sensors installed in the content area.

The proposed control board consists of a hardware part that operates multi-sensor signals and collects data from sensors, and a software part that transmits the collected data to the master board and controls the content module. The Figure 8 shows the components of the proposed control board as a block diagram[20].

Wireless communication consists of Wi-Fi and BLE, which are common wireless communication. communication technologies, are applied. The packet length can be adjusted from 8 bytes to a maximum of 32 bytes depending on the data amount of the interlocked sensor. Considering the increase in data amount, the communication packet speed was designed up to Mbps. In this study, a total of 16 sensor interfaces were configured, but since the sensor interface unit is controlled by linking each channel separately, the MCU was designed to increase the number of channels required.



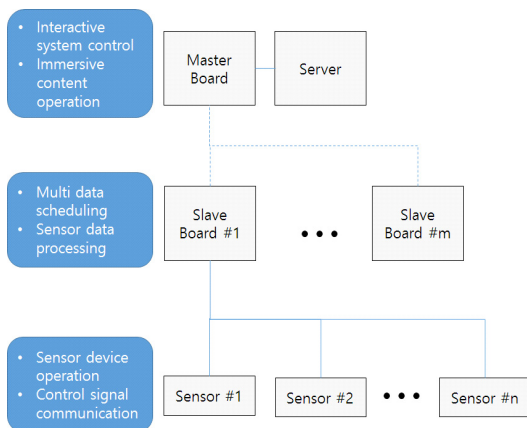
(Figure 8) Structure of the proposed control system

3.2.2 Command Packet

The data packet for transmitting the sensor control command of the master board to the slave board is 16 bytes in total, and the structure is Slave Addr(1 Byte), Header(2 Byte), CMD(2 Byte), Data(6 Byte), Checksum(3Byte), Tail (2 Byte). Header and tail packets were allocated 2 bytes each, Header: 0xAA/0x55, Tail: 0xEE/0xFF were fixed. In order to prevent errors from occurring due to input of the same packet in the middle of data, it was designed based on Timeout. In the packet time specification, the interval between packets is more than 5ms, and the communication interval is defined to be shorter than 5ms. When the slave board connects to the TCP/IP Server of the master board, the TCP/IP Server sends a live packet every 3 seconds. Alive CMD is 0xFF, and each slave board does not transmit a response packet, but each module sets a timer of 5 seconds when it receives an Alive packet. designed.

The data length of the packet transmitted from the slave board to the master board may require more bytes than 6bytes depending on the data size. Therefore, the data packet is designed in a non-fixed packet structure so that the length can be further increased.

It is designed to change the setting of the slave board according to the command sent from the master for simultaneous control of multiple multi-sensors, and the commands according to the CMD are shown in Table 4.



(Figure 7) Structure of the master-slave control system

(Table 3) Master-Slave Data Protocol

Master → Slave (16 byte/packet)					
Slave Addr. (1 Byte)	Header (2 Byte)	CMD (2 Byte)	Data (6 Byte)	Checksum (3 Byte)	Tail (2 Byte)
Addr	usable	0x00 ~ 0xFF (256 IDs available)			
Header	fixed	0xAA 0x55			
CMD	usable	0x00 ~ 0xFF (256 CMDs available)			
Data	usable	6 Bytes data payload			
Checksum	deterministic	{Slave Addr, CMD} + Data(6:4) + Data(3:1)			
Tail	fixed	0xEE 0xFF			

(Table 4) Slave-Master Data Protocol

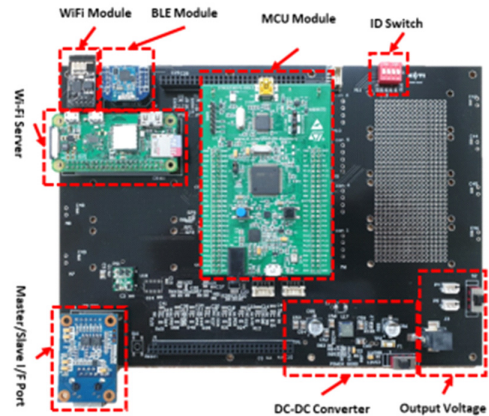
Slave → Master (variable size of packets)						
Slave Addr. (1Byte)	Header (2Byte)	CMD (2Byte)	Len (1Byte)	Data (Len Byte)	Checksum (8Byte)	Tail (4Byte)
Addr	deterministic	0x00 ~ 0xFF				
Header	fixed	0xAA 0x55				
CMD	usable	0x00 ~ 0xFF (256 CMDs available)				
Len	usable	Length of data payload				
Data	usable	Len bytes data payload				
Checksum	deterministic	4Bytes : {Slave Addr, CMD, Len} 4Bytes : pick and mix from Data				
Tail	fixed	0xFF 0xEE 0xEE 0xFF				

3.2.3 Control Board Implementation

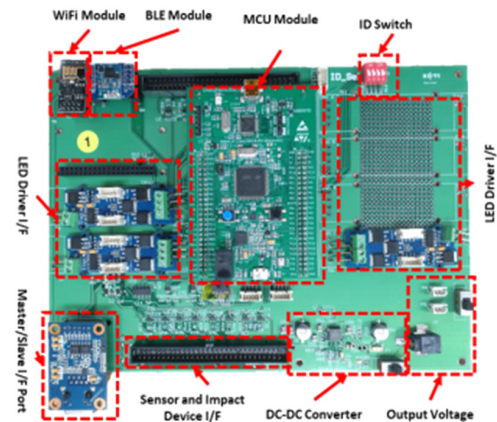
The control board provides various interfaces (UART, USB, GPIO, etc.). By using the 'STM32F4DISCOVERY' board that can be linked and controlled as a control system, it can be used universally even if the type of sensor used varies depending on the content. The Wi-Fi module and BLE module were applied for wireless interworking with the master board. The master board controls multiple slave boards at the same time. The 4-pin dip switch is installed so that the ID of each module can be easily changed for maintenance. In addition, the sensor interface is designed to be able to control 16 different sensors by configuring a 16 channels. The operating power is 12V/5A, and it includes a converter part for supplying power that meets the sensor specifications and an LED driver for controlling the sensor

operation including LEDs[21].

The firmware of the control board was produced by reflecting the operation algorithms such as simultaneous control of each sensor operation, data packet generation, and wireless transmission. A 16-channel interface test system was performed by interlocking multi-sensors, and the operation of each slave board was confirmed through master interworking.



(Figure 9) Master board overview



(Figure 10) Slave board overview

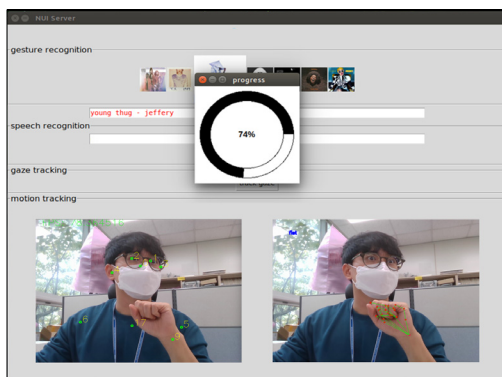
3.3 NUI Server

The NUI server is an application which reacts with human behavior using interaction data from the HC device manger and the LC device manger. NUI server get interaction data from the HC device manager via wired TCP

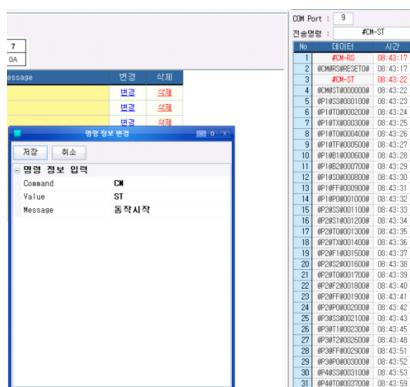
connection, and from the LC device manager via wireless Wi-Fi and BLE.

In this paper, we implement a simple example of NUI server to demonstrate if the proposed system works properly by user behavior such as pose estimation, hand pose recognition, and so on.

Figure 11 shows an example of controlling menu with user behavior. A user can move cursor by swiping hand right or left, and can select the menu by clenching one's fist. Also, the NUI server can get the text of user speech by clicking a button. Figure 12 shows the communication result of the LC device manager that received the multi-sensor data of the slave board. Depending on the operation, the data received from each sensor is checked along with the time information.



(Figure 11) An example of NUI server for menu control



(Figure 12) Simulation result of LC device command communication

4. Conclusion

An real-time multi-device control system has been proposed to support various type of sensors for NUI platform. The proposed system is consists of three sub-system: HC device manager, LC device manager, and NUI server. A HC device manager takes charge of high performance processing such as object detection, and deep learning. On the other hand, a LC device manager is considered to control multi-devices that require low computation, such as switches. Lastly, NUI server gathers user interaction data from the HC device manager and the LC device manager. In this paper, we demonstrate the proposed system properly operated. It is expected that it will be possible to adaptively install various types of interactive content by combining the two control systems according to the service environment. We will continue to study the proposed system to evaluate and improve the performance of the multi-device control system applied to commercial immersive content.

References

- [1] G. Lee, D. Shin and D. Shin, "NUI/NUX framework based on intuitive hand motion," *Journal of Internet Computing and Services*, vol. 15, no. 3, pp. 11-20, 2014. <https://doi.org/10.7472/jksii.2014.15.3.11>
- [2] T. Wu, K. Zheng, C. Wu and X. Wang, "User Identification Using Real Environmental Human Computer Interaction Behavior," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 6, pp. 3055-3073, 2019. <https://doi.org/10.3837/tiis.2019.06.016>
- [3] C. Yoon, C. Lee, S. Kwon, "A Study on the Interaction Smart Space Model in the Intact Environment," *Journal of the Korea Convergence Society*, vol.12, no. 1, pp. 89-97. 2021. <https://doi.org/10.15207/JKCS.2021.12.1.089>
- [4] Y. M. Park & W.T. Woo, "ARTable: AR based Interaction System using Tangible Objects," In *Proc. Korean Institute of Information Scientists and*

- Engineers (KIIE)*, pp.523-525. 2005.
https://doi.org/10.1007/11736639_150
- [5] L. Kim, H. Cho & S. Park, "SmartPuck System: Tangible Interface for Physical Manipulation of Digital Information," *Journal of KIIE*, vol. 34. no 4, pp.226-230. 2007.
<https://www.koreascience.or.kr/article/JAKO200734515985662.page>
- [6] J. Y. Han. "Low-cost multi-touch Sensing Through Frustrated Total Internal Reflection." In *Proc. ACM symposium on User Interface Software and Technology (UIST)*, New York, USA. Oct. 2005.
<https://doi.org/10.1145/1095034.1095054>
- [7] J. Rekimoto. "SmartSkin: an infrastructure for freehand manipulation on interactive surfaces." In *Proc. SIGCHI conference on Human factors in computing systems (CHI)*, Florence, Italy. Apr. 2002.
<https://doi.org/10.1145/503376.503397>
- [8] D. Vogel & R. Balakrishnan. "Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple users." In *Proc. ACM symposium on User Interface Software and Technology (UIST)*, Santa Fe, New Mexico, Oct. 2004.
<https://doi.org/10.1145/1029632.1029656>
- [9] C. Ganser, A. Steinemann and R. Hofer "Infractables: Supporting Collocated Group Work by Combining Pen-based and Tangible Interaction", In *Proc. IEEE. Horizontal Interactive Human-Computer Systems*, Newport, Rhode Island. Oct. 2007.
<https://doi.org/10.1109/TABLETOP.2007.38>
- [10] Balz Maag, Zimu Zhou, and Lothar Thiele. "A Survey on Sensor Calibration in Air Pollution Monitoring Deployments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4857-4870, Jul. 2018.
<https://doi.org/10.1109/JIOT.2018.2853660>
- [11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime multi-person 2d pose estimation using part affinity fields." In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*. Hawaii, USA, Jul. 2017. <https://doi.org/10.1109/CVPR.2017.143>
- [12] Bin Xiao, Haiping Wu, and Yichen Wei. "Simple Baselines for Human Pose Estimation and Tracking." In *Proc. European Conference on Computer Vision (ECCV)*. Munich, Germany, Sep. 2018.
<https://arxiv.org/pdf/1804.06208.pdf>
- [13] S. Jeong and D. Oh, "Development of a Hybrid Deep-Learning Model for the Human Activity Recognition based on the Wristband Accelerometer Signals," *Journal of Internet Computing and Services*, vol. 22, no. 3, pp. 9-16. 2021.
<https://doi.org/10.7472/jksii.2021.22.3.9>
- [14] S. Park, M. Ji and J. Chun, "2D Human Pose Estimation based on Object Detection using RGB-D information," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 800-816, 2018.
<https://doi.org/10.3837/tiis.2018.02.015>
- [15] C. Hao, Y. Wang, B. Jiang, S. Liu and Z. Yang, "Higher-Order Conditional Random Field established with CNNs for Video Object Segmentation," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 9, pp. 3204-3220, 2021.
<https://doi.org/10.3837/tiis.2021.09.007>
- [16] NVIDIA Jetson AGX Xavier Developer Kit. Available online:
<https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit>
- [17] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162-175, Jan. 2019.
<https://doi.org/10.1109/TPAMI.2017.2778103>
- [18] Marco Iorio, Massimo Reineri, Fulvio Rizzo, Riccardo Sisto, and Fulvio Valenza. "Securing SOME/IP for In-Vehicle Service Protection," *IEEE Trans. Vehicular Technology*, vol. 69, no. 11, pp. 13450-13466, Nov. 2020. <https://doi.org/10.1109/TVT.2020.3028880>
- [19] Douglas. C. Schmidt and Hans V. Hag. "Addressing the Challenges of Mission Critical Information Management in Next Generation Net-Centric Pub/Sub Systems with OpenSplice DDS," In *Proc. IEEE Int. Parallel Distrib. Process. Symposium (IPDPS)*, Miami, FL, USA, Apr. 2008.
<https://doi.org/10.1109/IPDPS.2008.4536567>

- [20] S. Chae, M. Kim, D. Lee and Y. Moon, "Implementation of Multi-Sensor Interface Module for Control User Interactive Content," *Information and Control System*, vol. 2019, no. 10, pp. 358-359, Oct. 2019.
<https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09262937>

- [21] M. Kim, S. Chae, M. Kim, and Y. Moon, "Real-time Multi-device Wireless Control System Implementation for Interactive Platform," *International Conference on Information Science and Technology (APIC-IST) 2021*, pp. 240-241, June, 2021.

● 저 자 소 개 ●



김 명 진(Myoung-jin Kim)

2015: Sejong University (Korea), Information and Communications Engineering (Bachelor Degree)

2017: Sejong University (Korea), Information and Communications Engineering (Master Degree)

2017~Present: Korea Electronics Technology Institute (Korea), Data Convergence Platform Research Center, Researcher

Research Interest: Interactive Platform, Wireless communication, User Interface

E-mail: kmjketi@keti.re.kr



황 태 민(Tae-min Hwang)

2015: Sogang University (Korea), Electronic Engineering (Bachelor Degree)

2017: Sogang University (Korea), Electronic Engineering (Master Degree)

2017~2021: LG Electronics (Korea), Senior Researcher

2021~Present: Korea Electronics Technology Institute (Korea), Data Convergence Platform Research Center, Senior Researcher

Research Interest: Natural User Interface, Edge AI, and Machine Learning

E-mail: taemin.hwang@keti.re.kr



채 승 훈(Sunghun Chae)

2012: Kookmin University (Korea), Electronics Engineering (Bachelor Degree)

2014: Kookmin University (Korea), Electronics Engineering (Master Degree)

2014~Present: Korea Electronics Technology Institute (Korea), Data Convergence Platform Research Center, Researcher

Research Interests: Embedded System, Interactive Platform, AR/VR, Data Convergence, Small Data etc.

E-mail: shchae@keti.re.kr

● 저 자 소 개 ●



김민준(Minjoon Kim)

2012: Yonsei University (Korea), Electrical and Electronic Engineering (Bachelor Degree)

2018: Yonsei University (Korea), Electrical and Electronic Engineering (Combined Master and Ph.D. Degree)

2018: Samsung Electronics, Korea, Engineer

2018~Present: Korea Electronics Technology Institute (Korea), Data Convergence Platform Research Center, Senior Researcher

Research Interests: SoC/VLSI design, FPGA Accelerator, Digital Signal Processing

E-mail: mjoon@keti.re.kr



문연국(Yeon-Kug Moon)

1998: In-ha University (Korea), Electronics Engineering (Bachelor Degree)

2000: In-ha University (Korea), Electronics Engineering (Master Degree)

2014: Korea University (Korea), Bio-microsystem Technology (Ph.D. Degree)

2001~2005: Samsung Electronics, System-LSI, Senior Engineer,

2005~Present: Korea Electronics Technology Institute (Korea), Data Convergence Platform Research Center, Director

Research Interests: Metaverse Platform, Data Convergence Platform Design

E-mail : ykmoon@keti.re.kr



김승준(SeungJun Kim)

1997: Korea Advanced Institute of Science and Technology (Korea), Electrical and Electronic Engineering (Bachelor Degree)

2000: Gwangju Institute of Science and Technology (Korea), Mechatronics (Master Degree)

2006: Gwangju Institute of Science and Technology (Korea), Mechatronics (Ph.D. Degree)

2006~2017: Carnegie Mellon University (US), Human-Computer Interaction Institute (post-doc, systems scientist)

2017~Present: Professor at Institute of Integrated Technology, Gwangju Institute of Science and Technology (GIST), Korea

Research Interests: Human-Computer Interaction (HCI), Extended Reality (XR), Artificial Intelligence (AI)

E-mail : seungjun@gist.ac.kr